REMARKS

The specification and claims 1-4, 7-13, 15, and 16 have been amended for clarification purposes only. Claim 17 has been added. Thus, claims 1-17 are currently pending in the case. Further examination and reconsideration of the presently claimed application are hereby respectfully requested.

Objection to the Claims

Claim 3 was objected to for an informality. In particular, the Office Action suggests that the grammatically improper and ambiguous language of claim 3 needs correction. To expedite prosecution, claim 3 has been amended in a manner that is believed to clarify the claim language. Accordingly, Applicants respectfully request that this objection be removed.

Section 112 Rejections

Claims 9 and 10 were rejected under 35 U.S.C. § 112, second paragraph, as being indefinite. To expedite prosecution, claims 9 and 10 have been amended in a manner that is believed to clarify the claim language. Accordingly, removal of this rejection is respectfully requested.

Section 103 Rejections

Claims 1, 3-10, and 12-16 were rejected under 35 U.S.C. § 103(a) as being unpatentable over a publication written by Matthew T. Nelson, entitled *Java Foundation Classes* (hereinafter "Nelson"). In addition, claims 2 and 11 were rejected under 35 U.S.C. §103(a) as being unpatentable over Nelson in view U.S. Patent No. 6,005,588 to Guha (hereinafter "Guha"). To establish a *prima facie* obviousness of a claimed invention, all claim limitations must be taught or suggested by the prior art. *In re Royka*, 490 F.2d 981, 180 U.S.P.Q. 580 (C.C.P.A. 1974), MPEP 2143.03. Obviousness cannot be established by combining or modifying the teachings of the prior art to produce the claimed invention, absent some teaching or suggestion or incentive to do so. *In re Bond*, 910 F. 2d 81, 834, 15 USPQ2d 1566, 1568 (Fed. Cir. 1990). The cited art does not teach or suggest all limitations of the currently pending claims, some distinctive limitations of which are set forth in more detail below.

None of the cited art teaches or suggests a first software component that is adapted to: (i) create a graphical representation of an object, which is embodied by code comprising text and other displayable content, and (ii) define visual attributes of the text without drawing the text. Amended independent claim 1 states, in part:

> A computer-readable memory medium, comprising: a first software component adapted to create a graphical representation of an object embodied as code within the software component, wherein the code comprises text and other displayable content... wherein the first software component is invoked during runtime by the application program to define visual attributes of the text, but not to draw the text.

Amended independent claims 8 and 16 recite similar limitations. It is noted that the amendments to claims 1, 8 and 16 are made for clarification purposes only, and should not be considered to be made for purposes of overcoming the cited art.

Generally speaking, the presently claimed case provides a system and method for overcoming a performance limitation of certain Swing Text Components. According to the presently claimed system and method, two lightweight peer components -- referred to as JTextFieldPeer and JFastLabelPeer -- may be created for implementing fast text drawing in an enhanced version of the Swing application program interface for Java applications. These lightweight peer components allow the Swing Text Components, JTextField and JLabel, to define the "look and feel" of text fields and labels that are displayed in the graphical user interface (GUI) of the Java application; however, the Swing Text Components (i.e., the first software components) are not allowed to draw the text. Instead, the peer components are used to invoke special fast text drawing code (i.e., embodied within the second software component) for drawing the text. Once the text is initially drawn, editing functions may be handled by the Swing Text Components. *See*, e.g., the Abstract of the Specification. Reference can also be made to page 31, line 24 to page 34, line 10 of the Specification for further description of the presently claimed system and method, according to one embodiment.

Nelson provides an overview of the Java Foundation Classes, which include Swing classes and others (Nelson, page xxv). Nelson, however, does not teach or suggest a first software component that is adapted: (i) to create a graphical representation of an object that is embodied by code comprising text and other displayable content, and (ii) to define visual attributes of the text without drawing the text. Statements in the Office Action suggest that teaching for the presently claimed first software component

may be found on pages 694, 697 and 78 of Nelson (*See*, e.g., Office Action, page 3). The Applicant respectfully disagrees, for at least the reasons set forth in more detail below.

On page 694, Nelson introduces the Swing JTextField Component and notes that the component may be used for getting simple text input and editing single-line text strings. Contrary to the statements in the Office Action, Applicants assert that the Swing JTextField Component cannot be used to provide teaching or suggestion for the presently claimed first software component. In other words, the JTextField Component cannot be used for defining the visual attributes of a text string without also drawing the text string. As noted on page 695 of Nelson, there are numerous ways in which the JTextField Component may be used to create a new text field (i.e., a graphical representation of an object). If a text string is not specified, the JTextField Component may be used to create a new text field without drawing any text. However, such a case is inconsistent with the presently claimed first software component, which specifically includes "text and other displayable content" within the object code of the first software component. In order for the JTextField Component to read upon the presently claimed first software component, there must be some teaching or suggestion within Nelson for not drawing text when a text string (e.g., "Horatio Hornblower") is specified for the JTextField Component. Such teaching or suggestion simply cannot be found within Nelson.

On page 697, Nelson describes how the border of the JTextField Component may be changed by using the "setBorder" and "setBackground" methods. Though the "setBorder" and "setBackground" methods may be used to change the "look" of the JTextField Component, the methods cannot be used (either separately, or in conjunction with the JTextField Component) to provide teaching or suggestion for the presently claimed first software component. For example, if the "setBorder" and "setBackground" methods are considered separately, the methods cannot be used to provide teaching or suggestion for the presently claimed first software component, since neither the "setBorder" nor the "setBackground" method is adapted to create a graphical representation of an object. Even if the "setBorder" and "setBackground" methods were combined with the JTextField Component, the combination of software components would still fail to provide teaching or suggestion for the presently claimed first software component. Since Nelson fails to teach (or even suggest) that the JTextField Component may refrain from drawing text when a text string is specified, and the "setBorder" and "setBackground" methods have absolutely nothing to do with drawing text, the software components cannot be combined to provide the presently claimed first software component and/or the functionality thereof. As a consequence, Nelson simply fails to teach or suggest all limitations of present claims 1, 8 and 16.

The Examiner suggests that "Nelson further teaches on page 78, UI class having separate groups of code to get the look-and-feel, and to draw the text." (Office Action, page 3). The Applicant agrees that separate groups of Swing-based code may be used for setting the look-and-feel of a displayed object and for drawing text within or upon the displayed object. For example, the "setLookAndFeel" method is commonly used to define the look-and-feel of a text field (and even the associated text) drawn by the JTextField Component. The Applicant recognizes the Examiner's intent to interpret the claims broadly. However, Applicant's assert that the claims cannot be interpreted so broadly to enable portions of the claims to be overlooked during prosecution. In determining the differences between the prior art and the claims, the question under 35 U.S.C. 103 is not whether the differences themselves would have been obvious, but whether the claimed invention as a whole would have been obvious. *Stratoflex, Inc. v. Aeroquip Corp.*, 713 F.2d 1530, 218 USPQ 871 (Fed. Cir. 1983); *Schenck v. Nortron Corp.*, 713 F.2d 782, 218 USPQ 698 (Fed. Cir. 1983); emphasis original; MPEP 2141.02. Distilling an invention down to the "gist" or "thrust" of an invention disregards the requirement of analyzing the subject matter "as a whole". *W.L. Gore & Associates, Inc. v. Garlock, Inc.*, 721 F.2d 1540, 220 USPQ 303 (Fed. Cir. 1983, cert. Denied, 469 U.S. 851 (1987); MPEP 2141.02. In providing such a loose interpretation of the presently claimed case, the Examiner fails to provide sufficient evidence within Nelson for the limitations actually recited in present claims 1, 8 and 16.

The Applicant also wishes to traverse, or otherwise correct, another statement that was made in the Office Action. On page 3 of the Office Action, the Examiner suggests that it "would have been obvious to one of ordinary skill in the art, having the teachings of Nelson that the JtextField and Jlabel components could be combined, to allow for one [of the] components to handle look-and-feel and one component to handle the displaying of the text." First of all, the presently claimed case does not attempt to combine the functionalities of the JTextField and JLabel components, nor can the claim language be interpreted as such. Rather, and as noted above, the presently claimed case allows the JTextField and JLabel components to separately define visual attributes of the text associated with the JTextField and JLabel components, however, the presently claimed case does not allow the JTextField and JLabel components to actually draw the text. Text drawing functionalities are provided by another software component, which may be utilized with either of the JTextField and JLabel components. The Applicant also points out that if the JTextField and JLabel components were combined, the combination would not only fail to read upon the present claims, but would also render the JTextField and JLabel components unsatisfactory for their intended purpose. Further clarification can be provided, if needed.

The secondary reference to Guha is not cited against present claims 1, 8 and 16. However, the Applicant wishes to point out that since Guha also fails to teach or suggest the above-mentioned limitation, Guha cannot be combined with Nelson to overcome the deficiencies therein.

For at least the reasons set forth above, none of the cited art, either separately or in combination provides motivation to teach or suggest all limitations of present claims 1, 8, and 16. Therefore, claims 1, 8, and 16, as well as claims dependent therefrom, are patentably distinct over the cited art. Accordingly, removal of this rejection is respectfully requested.

Patentability of the Added Claims

The present amendment adds claim 17 which is dependent from claim 1. Therefore, it is considered to be patentably distinct over the cited art for at least the same reasons as claim 1. Accordingly, allowance of added claim 17 is respectfully requested.

CONCLUSION

This response constitutes a complete response to all issues raised in the Office Action mailed February 13, 2004. In view of the remarks traversing the rejections, Applicants assert that pending claims 1-17 are in condition for allowance. If the Examiner has any questions, comments, or suggestions, the undersigned earnestly requests a telephone conference.

No fees are required for filing this amendment; however, the Commissioner is authorized to charge any additional fees which may be required, or credit any overpayment, to Conley Rose, P.C. Deposit Account No. 03-2769/5468-07500/.

Respectfully submitted,

Mollie E. Lettang

Mollie E. Lettang
Reg. No. 48,405
Agent for Applicant(s)

Conley Rose, P.C.
P.O. Box 684908
Austin, TX 78768-4908
Ph: (512) 476-1400
Date: May 13, 2004
JMF